FAULT DETECTION IN HVAC SYSTEMS USING MODEL-BASED FEEDFORWARD CONTROL

T. I. Salsbury, Ph.D.
Senior Research Engineer
Johnson Controls Inc.
507 E Michigan Street
Milwaukee, WI 53201, USA
Tel: 524-274-4660
Fax: 524-274-5810

R. C. Diamond, Ph.D.
Staff Scientist
Lawrence Berkeley National Laboratory
1 Cyclotron Road
Berkeley, California 94720, USA
Tel: 510-486-4459
Fax: 510-486-6658

ABSTRACT

This paper describes how a model-based, feedforward control scheme can improve control performance over traditional PID and detect faults in the controlled process. The scheme uses static simulation models of the system under control to generate feedforward control action, which supplements a conventional PID feedback loop. The feedforward action reduces the effect of plant non-linearity on control performance and provides more consistent disturbance rejection as operating conditions change. In addition to generating feedforward control action, the models act as a reference of correct operation. Faults that occur in the system under control cause the PID loop to provide a greater than normal control action to compensate for fault-induced discrepancies between the feedforward models and the controlled process. By monitoring the level of feedback compensation, faults can be detected in the controlled process. The paper presents results from testing the controller with a simulated dual-duct air-handling unit. We also discuss recent experiences of implementing the control scheme in a real building using the BACnet communication protocol.

KEYWORDS

Commercial Buildings, Comfort/Energy, Control/Simulation/Field Tests, Air-Conditioning Systems, Feedforward Control, Diagnostics

1 INTRODUCTION

Heating, ventilating, and air-conditioning (HVAC) systems are typically controlled using proportional plus integral (and sometimes plus derivative) PI(D) control law. In practice, HVAC systems exhibit non-linear operating characteristics, which causes control performance to vary when operating conditions change. Poor control performance can lead to occupant discomfort in the treated building, greater energy consumption, and increased wear on controlled elements, such as actuators, valves, and dampers.

In a conventional PI(D) feedback loop, the controller does not contain much information about the process it is controlling. Faults that lead to performance deterioration, or a change in system behavior, are often masked within a feedback loop. The control scheme described in this paper proposes the use of a feedforward controller as the primary controller, supplemented by a conventional PI(D) feedback loop. The model used in the feedforward controller acts as a baseline of correct behavior, and facilitates the detection of faults that develop in the controlled system. Incorporation of a system model in the feedforward control scheme reduces the impact of plant non-linearity, leading to more consistent control performance as operating conditions change.

Several researchers (e.g., Gertler, 1998; Glass et al., 1994; Isermann, 1995; Patton et al., 1995) have proposed fault-detection and diagnosis schemes based on the use of models. The main trade-off with model-based schemes is configuration effort versus model accuracy. Generally, the greater the potential accuracy of the models, the greater the effort required to configure the models for operation. We therefore selected models for the feedforward controller that are configurable from performance information typically available in design and commissioning records. Although these models may sacrifice some accuracy and fault detection sensitivity, we demonstrate that the proposed scheme can detect three important faults in the air-handling unit tested, namely reduced heating capacity, valve leakage, and a stuck return air damper.

2 DESCRIPTION OF THE CONTROLLER

Figure 1 shows the control and fault detection scheme. The simulation model, which is an *inverse* representation of the system generates a control signal, u_{FF}. An inverse model predicts the inputs to a system based on measured outputs. The signal u_{FF} is supplemented by a conventional PI(D) feedback loop, which generates control action u_{PI} based on the error between the setpoint and the controlled variable. The model is in static (steady-state) form and produces a control action appropriate for the current setpoint and measured disturbances. This control scheme is similar to one proposed by Hepworth and Dexter (1994), which used an adaptive neural network as the inverse system model.

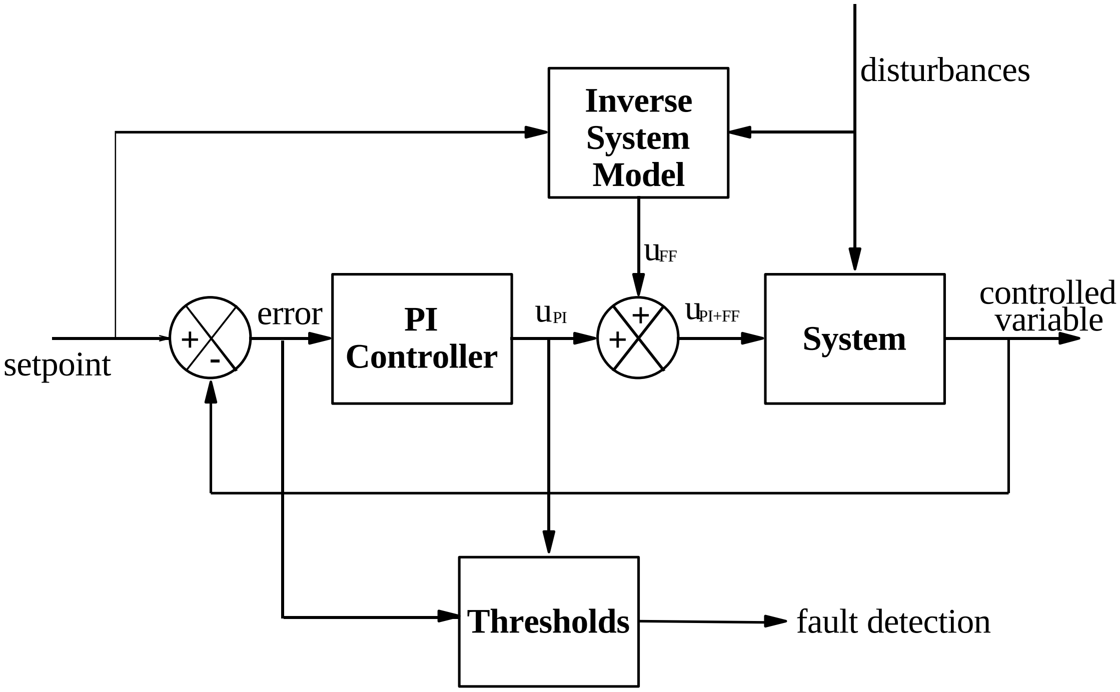


Figure 1: The control and fault detection scheme.

The inverse model, isolated from the feedback loop, produces responses based on the open-loop dynamics of the system. Note that because the model is a static model, the predicted control signal, u_{ff}, will change instantaneously for a change in any of the measured inputs. Maintaining the feedback loop speeds the response time of the controller and eliminate offsets resulting from model inaccuracies and unmeasured disturbances. Assuming the effect of *unmeasured* disturbances is small, the (steady-state) feedback control action (u_{PI}) serves as an indication of the model/system mismatch. The control action, u_{PI}, thus represents an implicit measure of the difference between the predicted and actual control signals for a particular setpoint. By configuring the model to represent a correctly operating system, the level of u_{PI} acts as an indication of fault development. Faults occurring in the system, which change its behavior or performance, create a mismatch between the model and system leading to an increase in feedback control action.

The control scheme incorporates fault detection capabilities by monitoring the magnitudes of two indicator variables. The first indictor variable is the output from the PI controller (u_{PI}) - the “control signal error” and the second is the difference between the setpoint and the controlled variable – the “setpoint error”. The controller generates an alarm if either of these variables exceeds a threshold for a sustained period.

The control signal error reveals changes caused by faults that do not affect the ability of the controller to maintain the setpoint, e.g., leakage through a control valve. A prolonged setpoint error that is not accompanied by a control signal error indicates a problem at or near to the point where the control signal would normally saturate, e.g., a capacity problem when full load is demanded. Simultaneous control signal and setpoint errors over a sustained period can indicate poor tuning or problems with the control loop. However, if the control loop is oscillatory, the errors may periodically return below their respective thresholds within a short enough time avoiding alarm generation. Sensor errors are also detectable by the control scheme. Those that do not affect the ability of the control scheme to achieve the setpoint will be detectable through the control signal error. Large errors in the controlled variable sensor that cause the setpoint to become unattainable would also be detectable through the setpoint error.

The control scheme triggers an alarm if the control signal error or setpoint error continuously exceed a threshold for a predetermined period. Figure 2 shows the fault detection algorithm. T_u is the threshold for the control signal error, T_e is the threshold for the setpoint error, and P is the maximum transgression period before generating an alarm. The fault detection part of the control scheme thus requires three parameters to configure it for operation: T_u, T_e, and P.

```
IF $|u_{PI}|>T_{\mathrm{u}}$ OR $|error|>T_{\mathrm{e},}$
        $P=P+Dt$
ELSE
        $P=0$
ENDIF
IF $P>P_{max}$
        $FAULT=1$
ELSE
        $FAULT=0$
ENDIF
```

Figure 2: Fault detection logic.

Under a PI control regime, the setpoint error is supposed to reach zero in steady-state. T_e, can thus be selected heuristically based only on considerations of typical sensor noise and tolerable tracking errors. The parameter, P_{max}, relates to the maximum time between periods of steady-state. For HVAC applications, it is reasonable to assume that transience does not normally persist for more than 30 minutes between periods of (quasi) steady-state. We thus selected a value of 30 minutes for P. Selection of the threshold T_u is more difficult and relates to the accuracy of the models and the degree of detection sensitivity required. Ideally, T_u should be established through tests on the correctly operating system. However, as is shown in Section 6, T_u may be also be set heuristically for preliminary testing in order to detect gross faults in the system.

3 MODELS USED IN THE FEEDFORWARD CONTROLLER

In this paper, we apply the controller to a dual-duct air-handling unit, which has heating, cooling, and mixing capabilities. The controller incorporates three separate models: mixing box, heating coil, and cooling coil. Details of the model equations can be found in (Salsbury, 1998). The models in the feedforward controller are simplified in several respects. In particular, they do not treat:

- Variations in coil thermal conductance with fluid flow rates;
- Dehumidification in the cooling process;
- Valve/damper non-linearity

We make the latter simplification because characterization of this non-linearity requires parameters that are not easily obtainable or reliable, such as the inherent and installed characteristics of the valves and dampers. The simplification is reasonable, as one of the goals of the design and commissioning processes is to linearize the relationship between the control signal and controlled variable, e.g., by canceling coil non-linearity with valve non-linearity. Although the model simplifications reduce potential accuracy and performance of the scheme, a major advantage is that the parameter values may be obtained from typically available information, rather than requiring calibration data and additional tuning effort.

Table 1: Configuration parameters

PARAMETER/DESIGN SPECIFICATIONS	UNITS
HEATING/COOLING COIL	
Heat transfer rate	kW
Cold fluid inlet air temperature	°C
Cold fluid mass flow rate	kgs^{-1}
Hot fluid inlet temperature	°C
Hot fluid mass flow rate	kgs^{-1}
MIXING BOX	
Minimum fractional outside air flow	%

Table 1 lists the parameters required by the models in the feedforward controller and Table 2 lists the required sensor measurements/variables. Note that since the considered air-handler is a dual-duct system, air temperatures and flow rates are required before the coils in both the hot and cold ducts.

Table 2: Required sensor signals/variables

SENSOR SIGNAL	UNITS
Return air temperature	°C
Outside air temperature	°C
Air flow rates (hot and cold ducts)	kgs^{-1}
Pre-coil air temperatures (mixed air)	°C
Setpoints (mixed, hot-air, cold-air)	°C

4 SIMULATION TEST SYSTEM

The control scheme is evaluated using a simulation of an air-handling unit developed in the MATLAB environment using component models adapted from HVACSIM+ (Clark, 1985). The simulation replicates a real system installed in a large (1.4 million square feet) federal building in downtown San Francisco, USA. Figure 3 depicts the air-handling unit, which is a dual-duct type having three thermal subsystems: mixing box, cooling coil, and heating coil. The air-handling unit has the capacity to deliver 74kg/s of air and provide 850kW of heating and 1260kW of cooling.

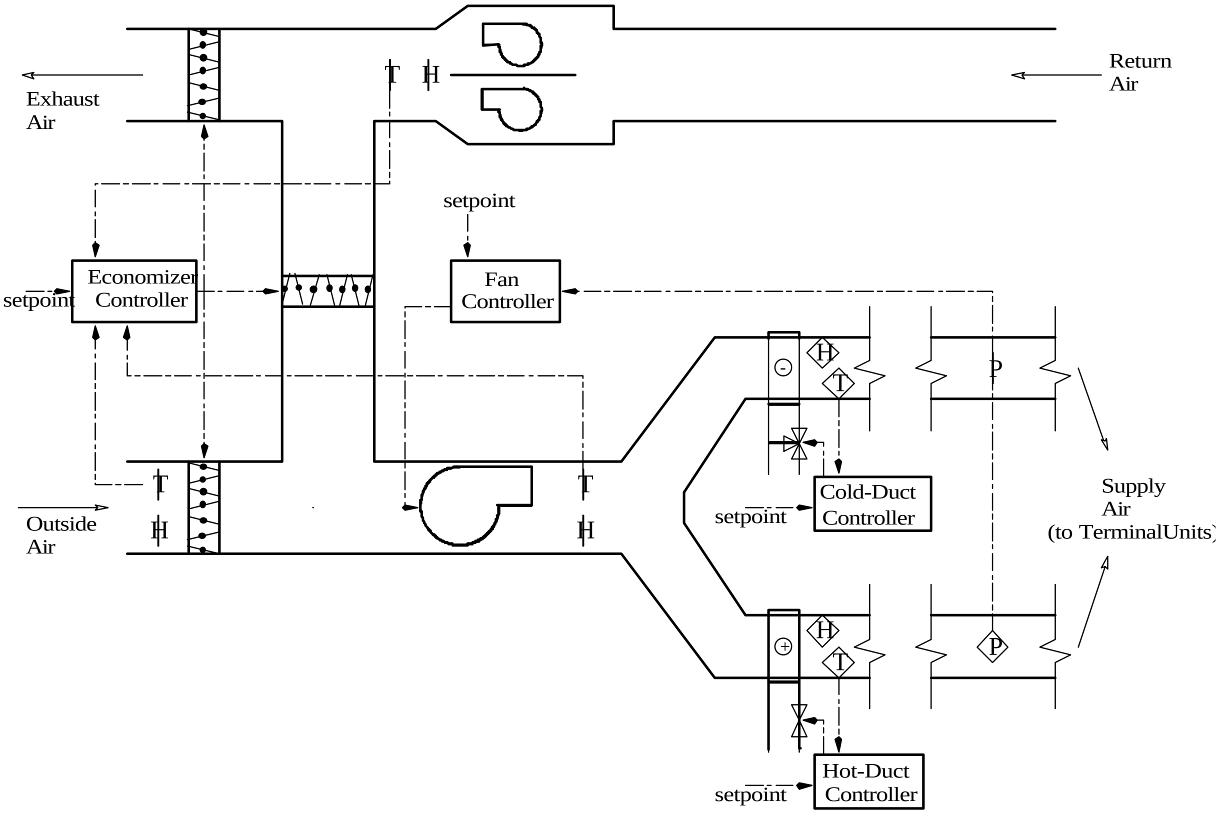


Figure 3: Schematic of the dual-duct air-handling unit.

Each thermal subsystem has its own controller. The mixing box controller modulates three sets of dampers to maintain mixed air conditions. There is a minimum outside-air requirement based on damper position (20% minimum outside-air) and a temperature-based economizer. The hot duct houses a steam-to-air heating coil regulated by a two-way valve, and there is a water-to-air cooling coil having a three-way valve in the cold duct. The fan speed varies according to load changes in the zones in a conventional VAV arrangement to maintain constant static pressure in the supply ducts.

5 SIMULATION TEST RESULTS

We evaluated the control scheme in two respects:

- Setpoint tracking performance;
- Ability to detect faults in the three thermal subsystems.

Twelve hours of test (simulated) data, sampled at 5-second intervals were used to evaluate the control scheme. The test data contain real measurements of ambient and return air temperatures from the building. We varied the supply airflow rate in a sequence of arbitrary steps and ramps in order to provide excitation similar to VAV control. Figure 4 shows the variation in return and ambient air temperatures in addition to the variation in airflow (at the supply fan) during the test data. Note that the supply airflow split between the hot and cold ducts. We also varied the setpoints for each of the three subsystems to facilitate a rigorous assessment of the controllers.

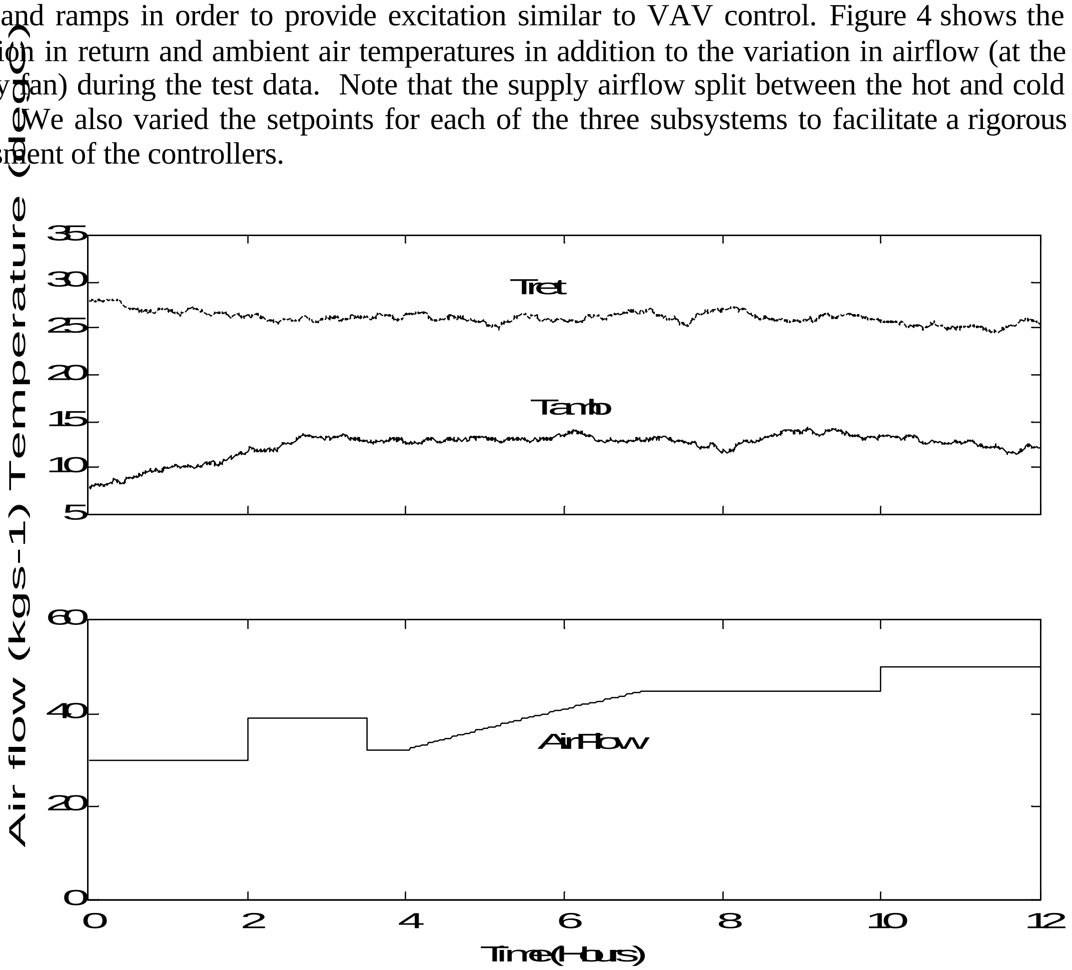


Figure 4: Disturbances other than setpoint changes during the test period. Note that airflow is at the supply fan, i.e., before the duct splits.

5.1 Control Performance

This section compares the performance of the feedforward controller with that of a conventional PI controller. We tuned each of the PI control loops using a standard open loop tuning rule (Zeigler-Nichols) based on the highest gains exhibited in the range of conditions in the test data.

Figure 5 shows the simulated performance of the PI controller when subjected to the test data. The upper graph shows the three controlled variables and setpoints and the lower graph shows the control signals to each of the subsystems. For these conditions, the PI controller provides satisfactory performance for the test data. We characterized control performance by calculating the sum of the mean absolute errors (MAE), listed in Table 3.

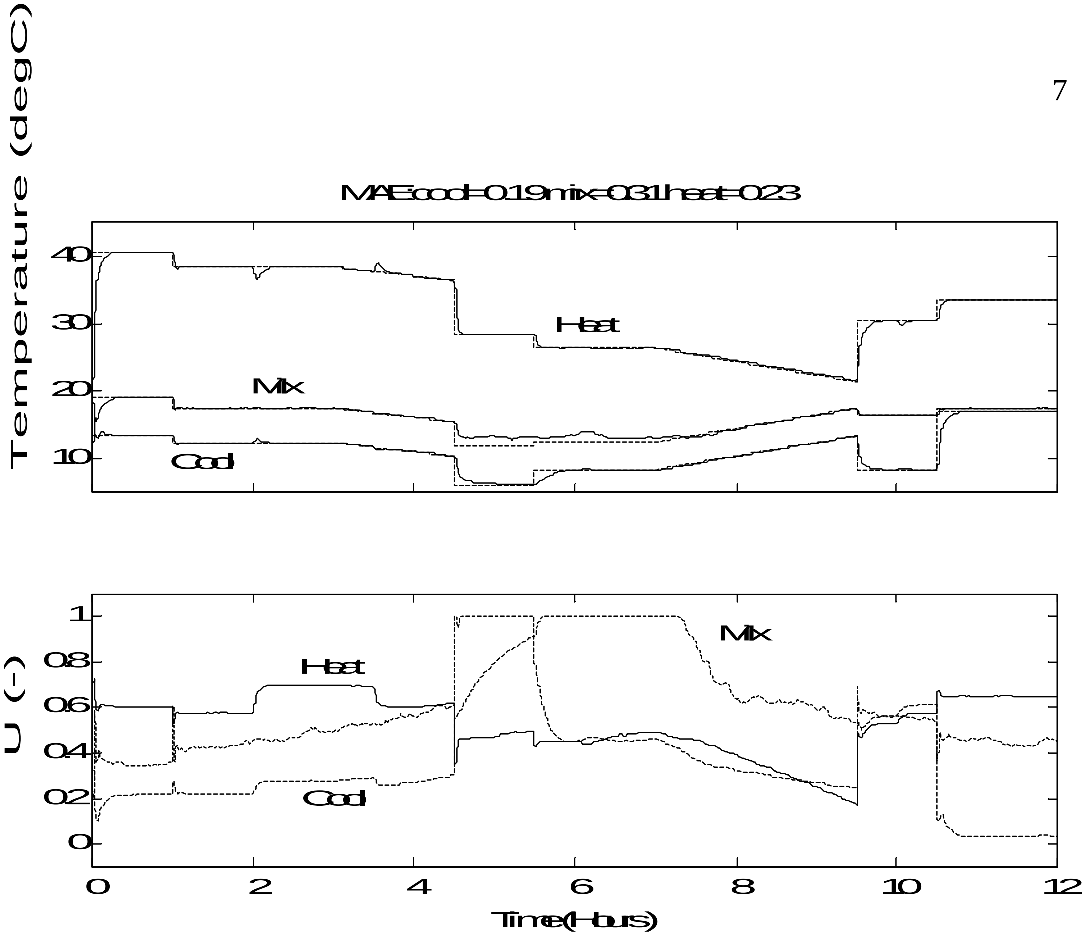


Figure 5: Control performance with PI control.

Figure 6 shows the performance of the same PI controllers, but this time with the inclusion of supplemental feedforward action. The graphs show improvements in the responses to the disturbances, particularly for the heating coil control-loop, which exhibits tighter control toward the end of the test data.

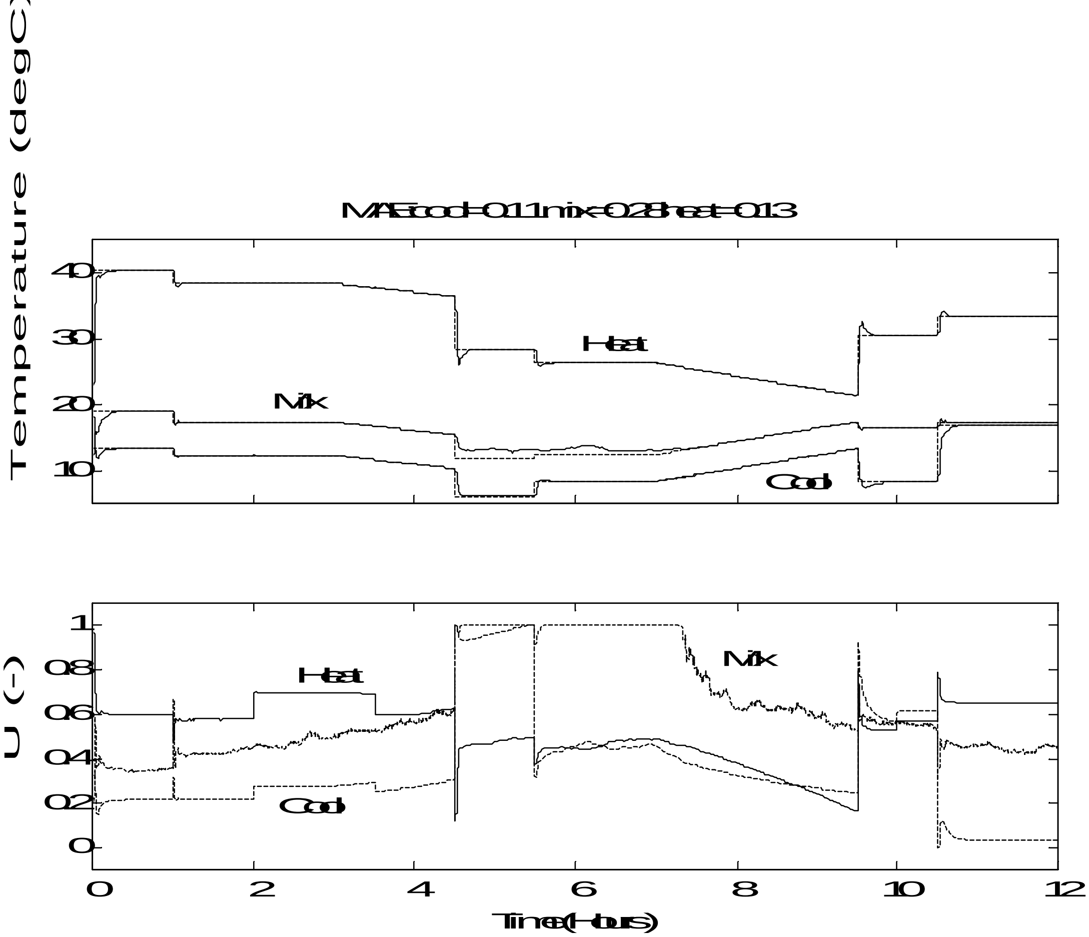


Figure 6: Control performance with feedforward control.

Table 3 lists the mean absolute errors in each control loop over the period of the data. The smallest improvement in performance is in the mixing control-loop, which shows only a 10% reduction in the MAE. This is due to the ambient and return air temperatures not varying significantly during the test data, leading to a reasonably invariant gain in this subsystem. In practice, variations in ambient air temperature significantly affect the mixing box gain and lead to instability in this loop. Previous work has shown that the feedforward controller can significantly improve the mixing box control-loop in the face of highly variable environmental conditions (Salsbury, 1998). The cooling coil control performance improves by 42%, while the heating coil shows a slightly greater improvement of 43%.

Table 3: Comparative performance of controllers

	MAE (K)		
PROCESS	COOL	MIX	HEAT
PI control in isolation	0.19	0.31	0.23
PI + feedforward control	0.11	0.28	0.13
Improvement (%)	42	10	43

5.2 Fault Detection Performance

In this section, we describe tests to evaluate the ability of the control scheme to detect faults in the three thermal subsystems. Note that we used the same test data as in the control performance tests.

5.2.1 No Fault Condition

First, we tested the control scheme with the system in its correctly operating condition. Figure 7 shows the indices associated with the fault detection. The top graph shows the control signals generated by the three feedback control loops in the air-handler (heat – solid line; cool - dashed line; mixing – dash-dot line). Recall that the PI control signal represents an implicit measure of the difference between the predicted and actual control signals for a particular setpoint. If the models used in the controller were perfect representations of the system under control, these control signals would asymptotically approach zero following each disturbance.

Figure 7 shows that there are modeling errors in each of the models. The errors vary during the data due to transient effects and because the models approximate the system better in certain parts of the operating range than in others. The cooling coil exhibits the greatest level of inaccuracy (e.g., around 10 hours), this being most likely due to the extra simplifications in this model, such as non-treatment of dehumidification. The heating coil model appears to represent the system well, with the feedback contribution being the lowest of the three subsystems. The bottom three graphs in Figure 7 show the fault detection output for each loop (1 for presence of a fault, 0 for no fault). A fault is deemed to have occurred if the feedback control signal rises above the threshold and stays above it for a sustained period (set to 30 minutes for these tests).

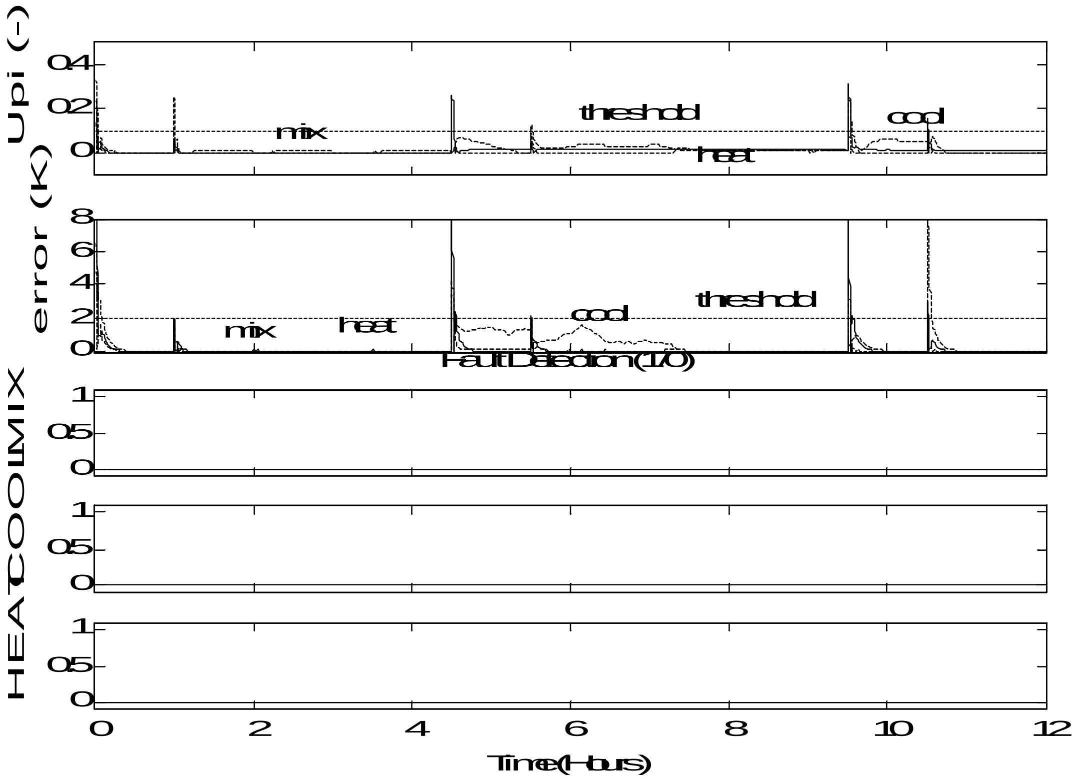


Figure 7: Fault detection indicator variables – correctly operating system.

The selection of a threshold is primary to the performance of the fault detection scheme. We selected a threshold using the correct operation data based on a 99% confidence interval determined for each of the indicator variables. For simplicity, we calculated a single threshold for all three subsystems. In practice, individual thresholds could be obtained by performing closed-loop tests on each of the subsystems during the commissioning process. These tests could be set up so that selected setpoints exercise the systems at strategic points in their operating ranges. Alternatively, thresholds could be estimated during an on-line learning phase, whereby maximum steady-state feedback action is monitored over an assigned period. Alternatively, if empirical determination is not possible, default threshold values could be determined for particular classes of HVAC systems based on statistically significant tests on similar equipment.

5.2.2 Reduced Heating Capacity Fault

Using the thresholds established from the correct operation data, we tested the fault detection capability of the controller under the condition of a 50% reduction in the capacity of the heating coil. We altered the heat transfer rate in simulated system by changing the UA of the coil. In practice, a reduction in heating capacity can occur for a number of reasons, such as: coil fouling, reductions in boiler efficiency, reductions in steam flow, etc.

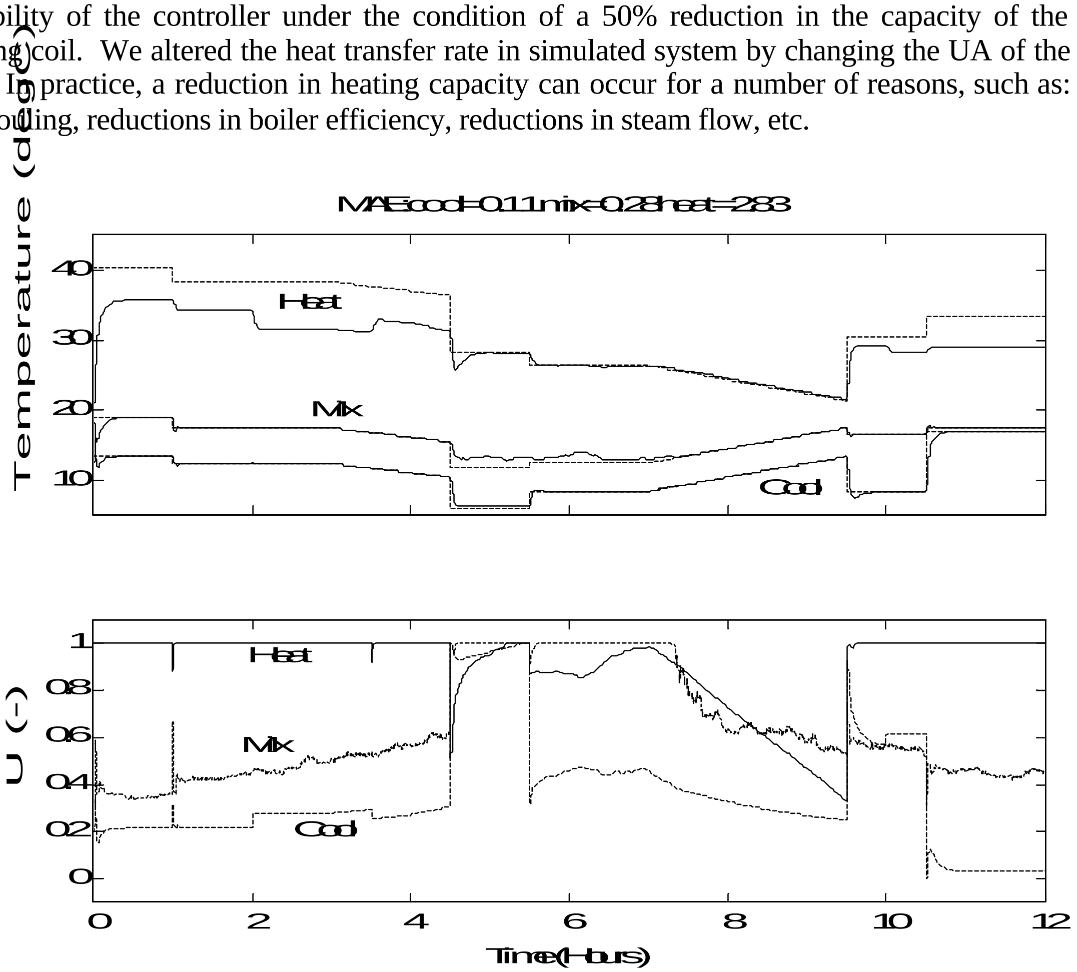


Figure 8: Control performance – reduced heat capacity fault.

Figure 8 shows the behavior of the controller with the reduced-capacity heating coil. The upper graph in the figure shows that the setpoint is unattainable during certain periods in the data, due to the effect of the fault. The heating coil control signal saturates for much of the data at 100%, corresponding to a fully open valve. Figure 9 shows the fault detection indicators in the presence of the reduced heating capacity fault. The fault is evident in both the control signal error and the controlled variable error. The fault is detectable from the control signal error for the entire data set, but masked from the setpoint error at certain times, e.g., during the middle of the test day. The software triggers an alarm after 30 minutes into the data set and sustains the alarm throughout the remaining data.

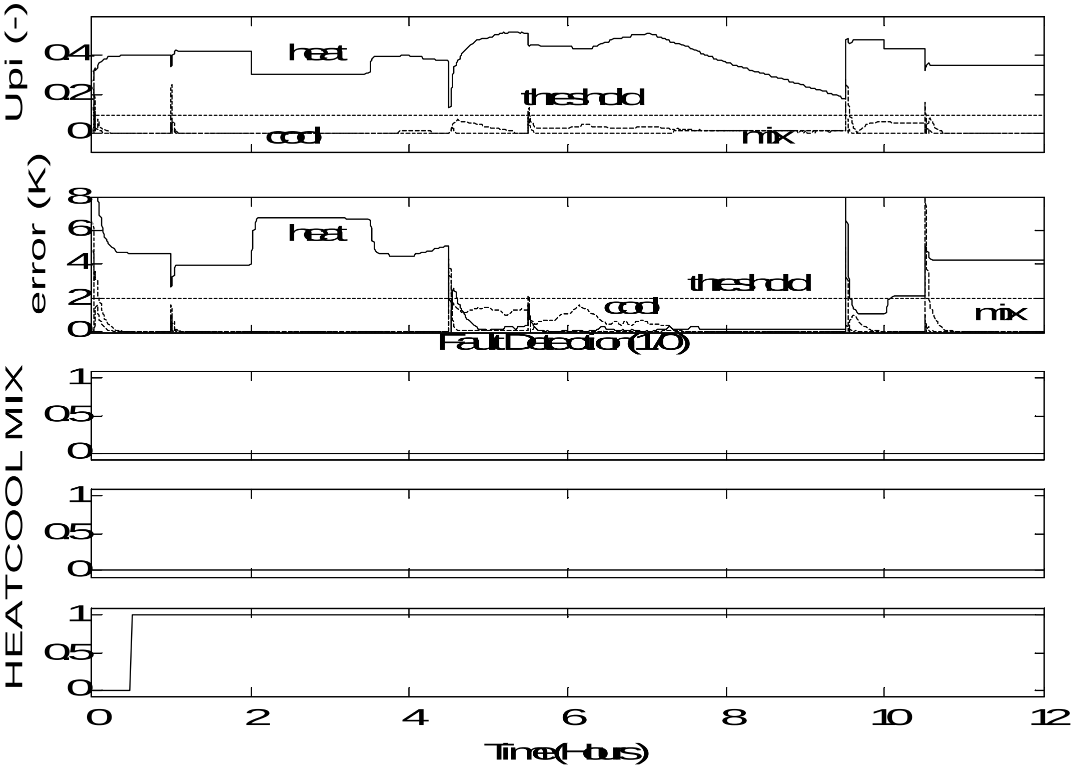


Figure 9: Fault indicator variables – heating capacity fault.

5.2.3 Valve Leakage

Leakage through the control valve of a heat exchanger is a common fault in HVAC systems, but often persists unnoticed due to feedback masking and compensatory actions of other systems. This fault can lead to wastage of energy, not only in the subsystem where the fault occurs, but also in other downstream systems that provide compensatory action. In this test, we set up the cooling coil valve model in the simulation so that the closed valve leaked by 20% of maximum flow. Figure 10 shows the behavior of the control scheme for the leakage fault. The 20% leakage causes the cooling coil to provide too much cooling during most of the test data, which makes the feedback loop saturate the control signal at zero. The controller only attains the setpoint for a higher cooling demand, e.g., during the middle of the data.

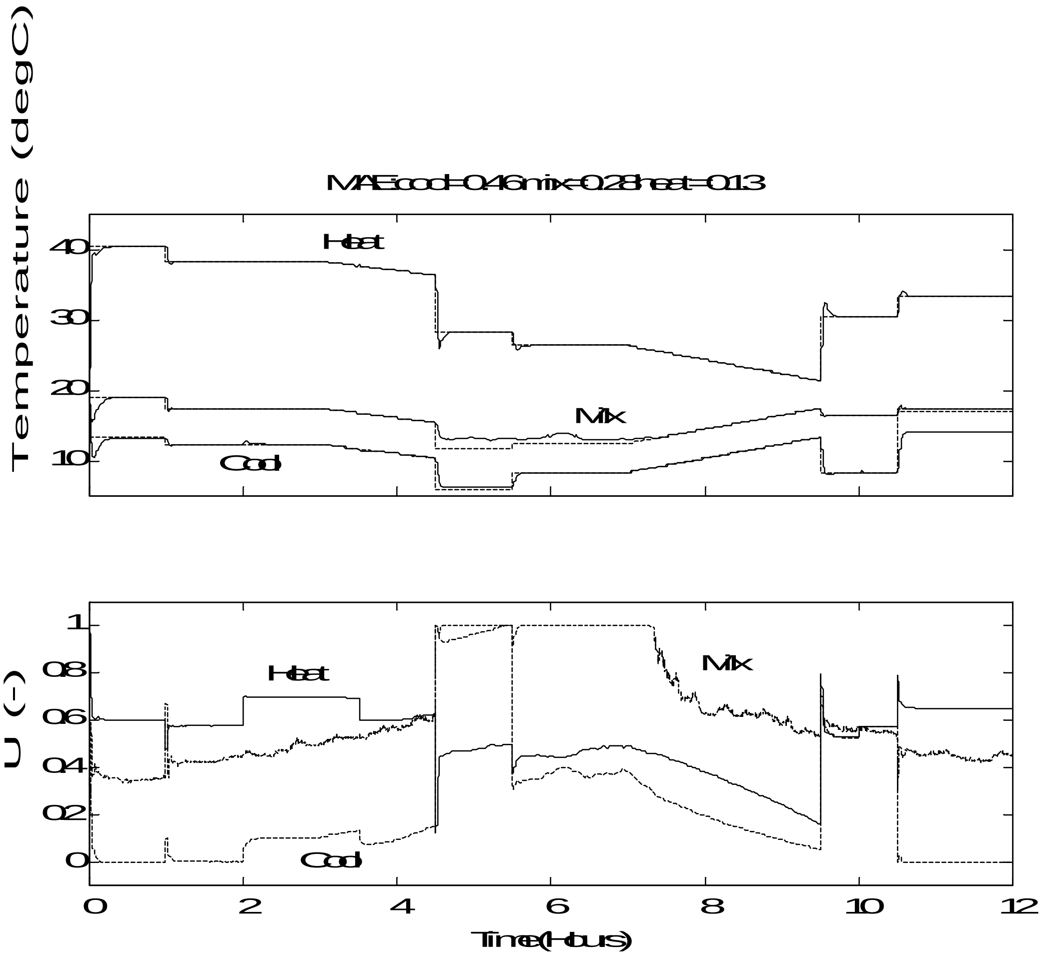


Figure 10: Control Signals and Temperatures – Leaking Cooling Valve Fault.

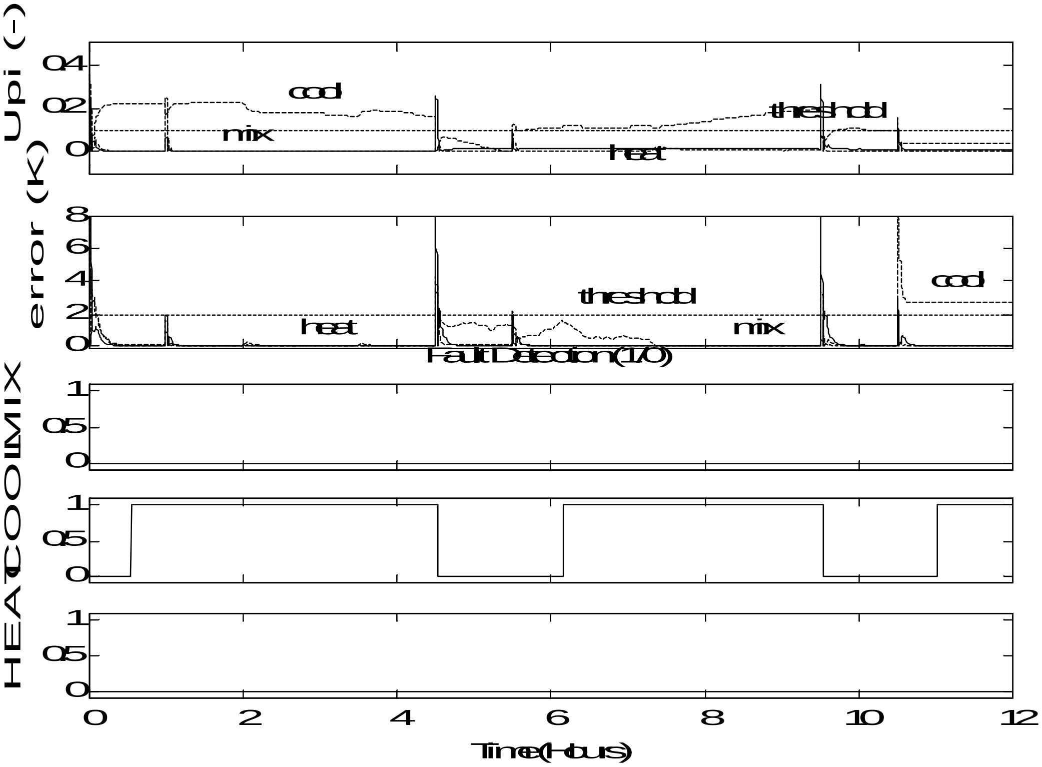


Figure 11: Fault Indicator Variables – Leaking Cooling Valve Fault.

Figure 11 shows the fault detection indicators for the leakage fault case. The top graph shows that the feedback control action exceeds the threshold for a large portion of the day. Correspondingly, the cooling fault indicator graph (second from bottom in the figure) indicates a fault for much of the data. The control signal error is largest at low load regions due to this being the part of the operating range where the fault is most evident. When the coil operates at a higher load, the feedback compensation drops below the threshold. The scheme therefore only detects faults when the considered subsystem is operating at a point where the effects of the fault are evident. Note that toward the end of the data, control signal saturation masks the fault on the control-signal error indicator. However, the leaking valve makes the setpoint unattainable leading to an alarm signal.

5.2.4 Stuck Return Air Damper

A stuck damper in the mixing box is another common fault in HVAC systems. This fault can be due to various causes, such as a failed actuator, damper obstruction, de-coupled linkage, etc. The fault is difficult to detect in practice as it does not cause complete failure of the mixing process, but instead alters its characteristics and restricts its operating range. Figure 12 shows the performance of the controller when the re-circulation damper is stuck at 50% open. The feedback loop masks the effect of this fault at the beginning and end of the day by maintaining the setpoint quite well. However, control performance deteriorates in the middle of the day when the controller demands 100% outside air.

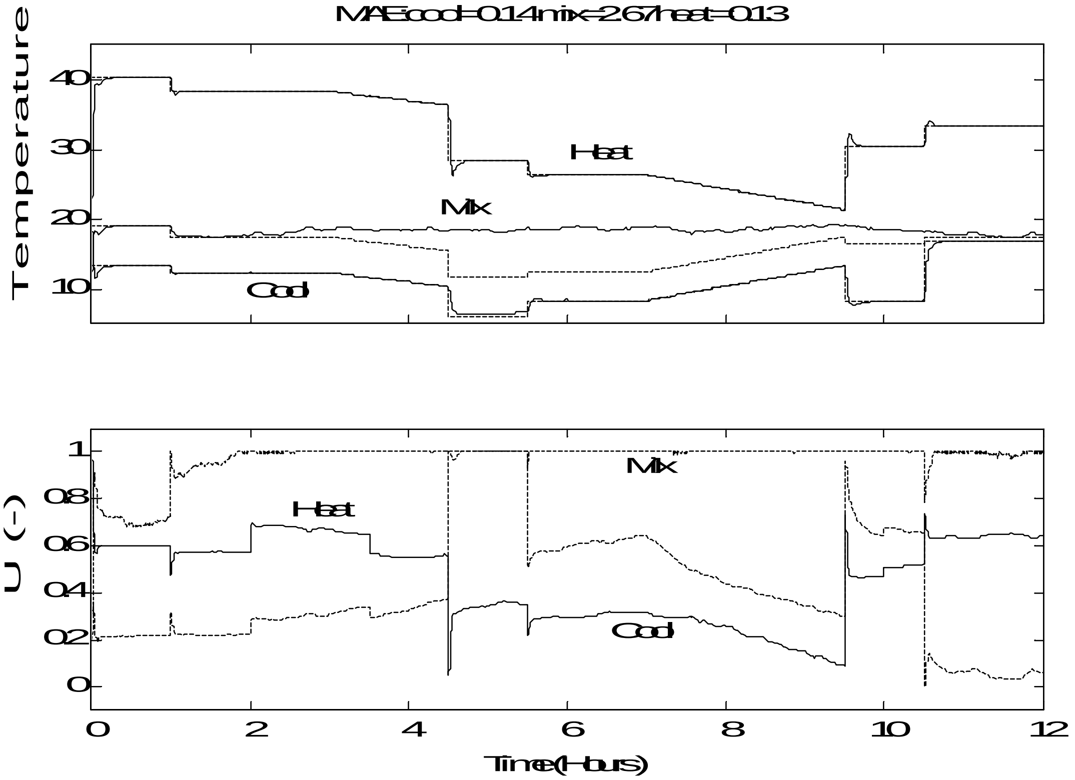


Figure 12: Control performance – stuck mixing damper.

Figure 13 shows the fault indicator variables for the stuck damper condition. The feedback control action to the mixing box indicates the fault when the effect is masked in Figure 12. During the middle part of the day, unwanted re-circulation through the mixing box reduces its operating range so that setpoints close to the ambient air temperature become unattainable. The combination of the control signal error and the controlled variable error allow detection of the fault during all the data, with the exception of the first 30 minutes (which was the time set for P_{max}).

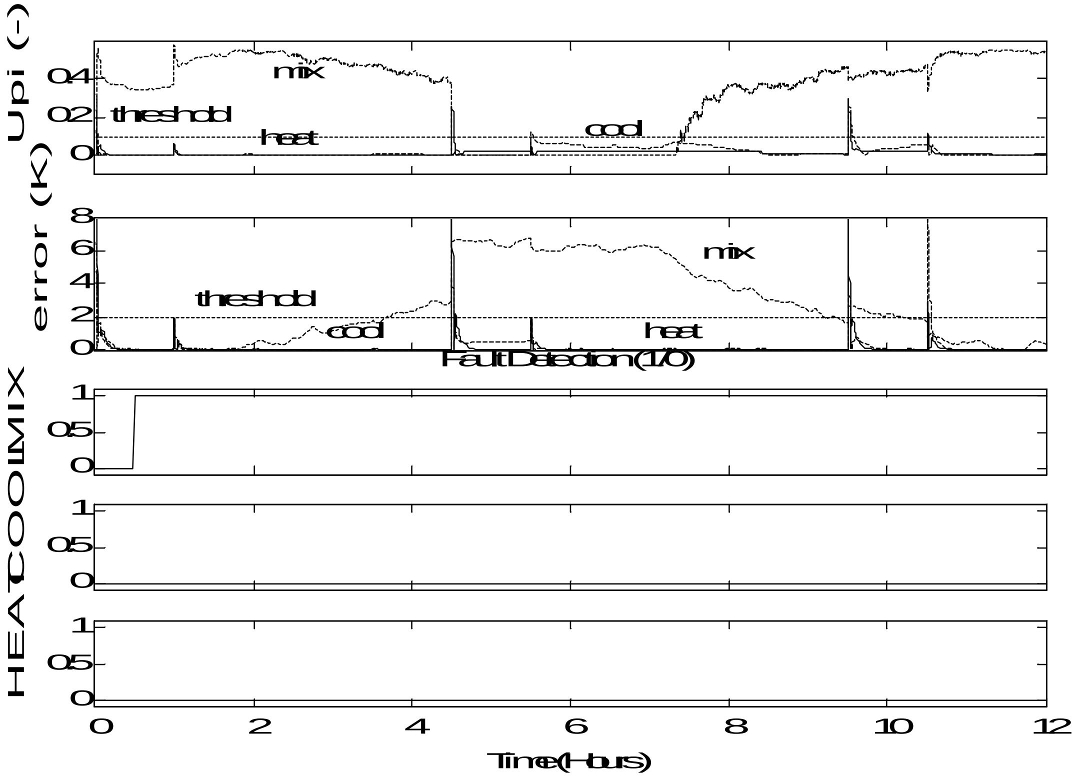


Figure 13: Fault indicator variables – stuck mixing damper.

6 IMPLEMENTATION IN A REAL BUILDING

We developed the control and diagnostics algorithms into a stand-alone software program for testing in a real building on a dual-duct air-handler that formed the basis of the simulated system described in Section 4. We initially deployed the tool in a passive mode with the intention of validating the models and establishing thresholds.

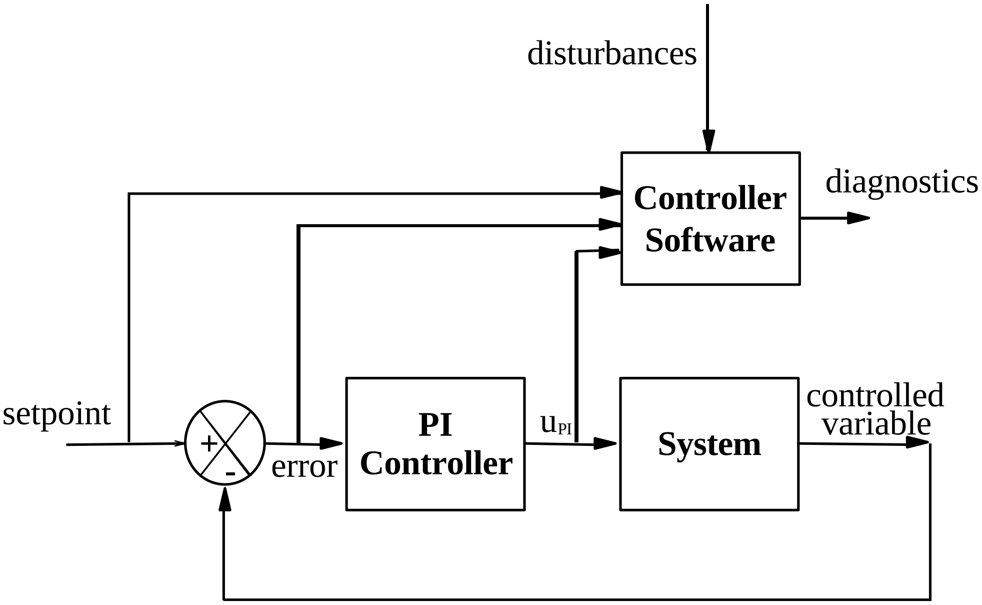


Figure 14. Interaction of Control Software with PI-Loop When in Passive Mode

Figure 14 depicts how the controller software was set up to interact with a PI loop in passive mode. In this mode, the feedforward control signals generated by the models do not affect the control operation and the system remains under PI-only control. In terms of fault detection, instead of using the PI control signal (u_{PI}) as a measure of the difference between the predicted and actual control signals, the difference is calculated explicitly, i.e., u_{PI} - u_{FF}.

6.1 Software Architecture and Connection to the EMCS

We developed the software based on three separate modules, as shown in Figure 15. The user interface provides diagnostic information to the user and allows the user to change parameters of the feedforward models, and other configuration information. The central module contains the control and diagnostics algorithms that function according to configuration information set by the user and data obtained from the energy management and control system (EMCS) network. The third module (control system interface) handles acquisition of data from the EMCS. The building in which we performed the tests was the subject of a recent large-scale EMCS retrofit, which included replacing a large part of the system with BACnet (ASHRAE, 1995) compliant control devices. We thus developed the control system interface to use the BACnet communication protocol. Use of this communication protocol opens the way for testing the control software on any other BACnet compliant system regardless of the manufacturer.

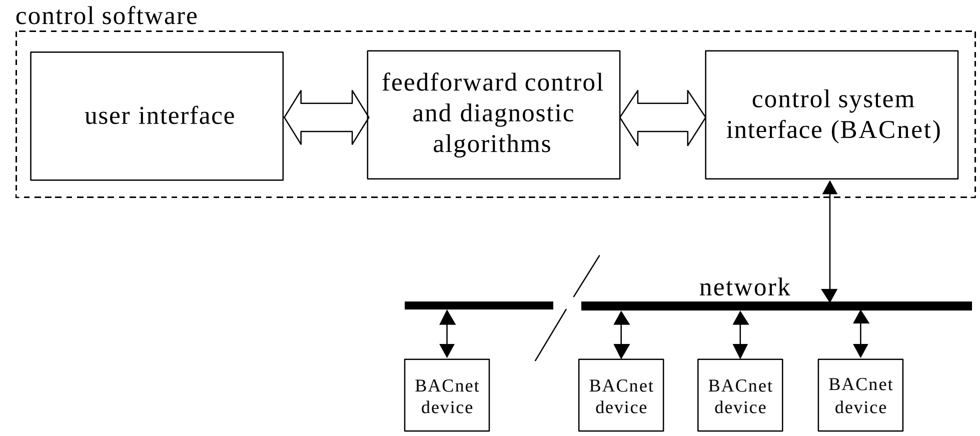
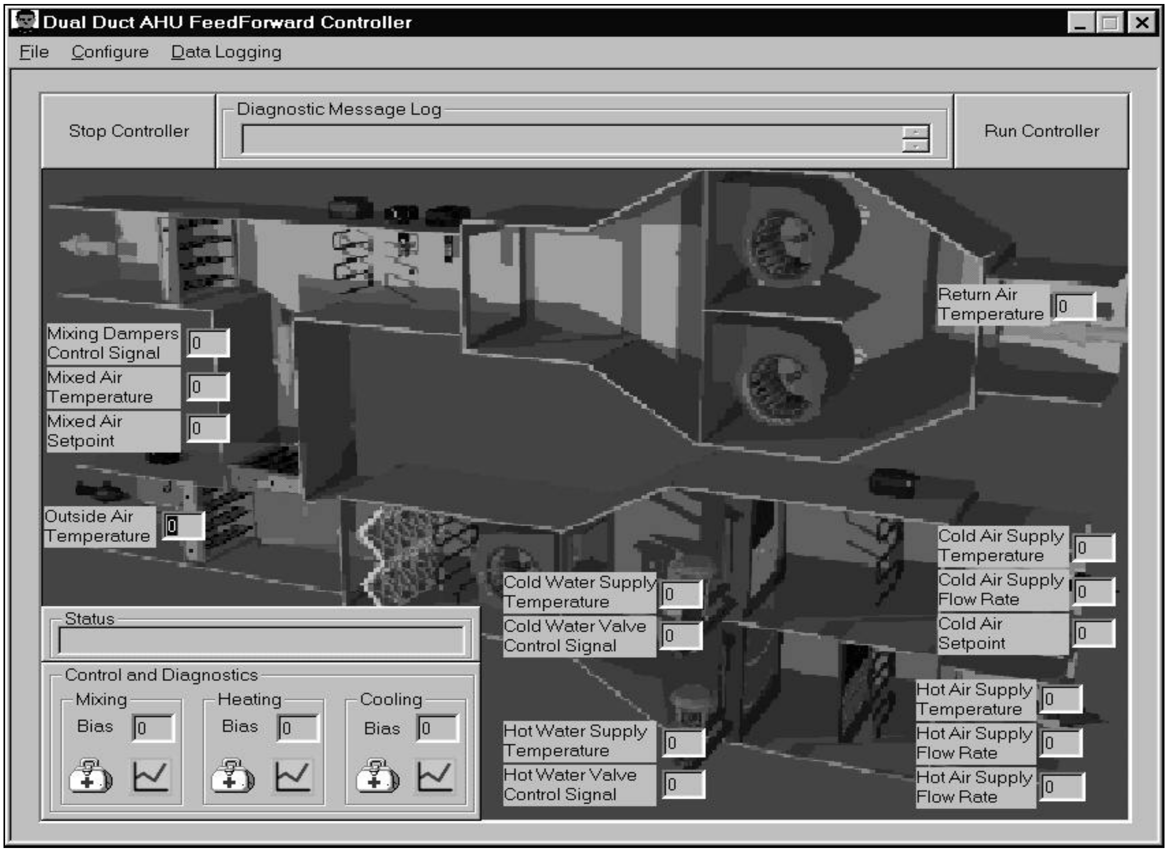


Figure 15. Software Module Interaction and Connection to the Control System

Figure 16 shows the user interface, which depicts the dual-duct air-handler used for the tests. Note that the two fans in the return duct have their speeds tracked to the speed of the supply fan, which is regulated in order to maintain the average of the hot- and cold-duct static pressures at a setpoint.



Return Air Temperature 0

Mixing Dampers Control Signal 0

Mixed Air Temperature 0

Mixed Air Setpoint 0

Outside Air Temperature 0

Cold Air Supply Temperature 0

Cold Air Supply Flow Rate 0

Cold Air Setpoint 0

Cold Water Supply Temperature 0

Cold Water Valve Control Signal 0

Hot Air Supply Temperature 0

Hot Air Supply Flow Rate 0

Hot Air Supply Flow Rate 0

Hot Water Supply Temperature 0

Hot Water Valve Control Signal 0

Mixing Bias 0

Heating Bias 0

Cooling Bias 0

Figure 16. User Interface Showing the Dual-Duct Test Unit

6.2 Obtaining a Points List

Before we could carry out the tests, we had to obtain a point list of the sensors and control signals required by the software tool. This task is unavoidable for any application that needs to poll data from an EMCS network. The devices on the network that relate to the physical sensor and control-signal measurements required by the application require identification so they can be mapped onto the variables in the application program. The process of acquiring the necessary information can be both time consuming and subject to human error.

6.3 Sensor Availability and Accuracy

One problem encountered during the testing of the feedforward controller concerned sensor availability. In the test system, direct measurements of airflow in the hot and cold ducts were not available. The feedforward models use these measurements to calculate temperature rises/drops across the coils and the model predictions are quite sensitive to these variables. We therefore had to proxy the air flow rates using other sensor measurements that were available. We applied simple models to calculate airflow from the supply fan VFD control signal and static pressure measurements in the hot and cold ducts. The proxy was difficult to assess for accuracy, as we were only able to obtain point measurements of actual airflow at sporadic operating points.

6.4 Initial Test Results

The original aim of the first phase of testing was to validate the models to establish the threshold values, T_u. However, it became apparent in the early stages of testing that blindly using data from the system in its “normal operation” state to set thresholds was inappropriate. We found that normal operation did not necessarily mean “correct operation”. The initial test described in this section therefore entailed detecting pre-existing faults in the system. We discovered that the tool was useful as a re-commissioning aid and could be used in this way by setting T_u heuristically before carrying out the tests.

As explained earlier, the software operated in passive mode and maintained its fault detection capability by calculating the difference between the feedforward control signal and the measured PI control signal explicitly. During the test, the supply fan-speed and static pressures remained relatively constant, which reduced the effect of errors stemming from the airflow proxy. In addition, the return and ambient air temperatures did not vary significantly during the tests. The effect on the AHU behavior from variations in measured disturbances was therefore small during the test period. Figure 17 shows the return and ambient air temperatures and the airflow proxy in the hot and cold ducts.

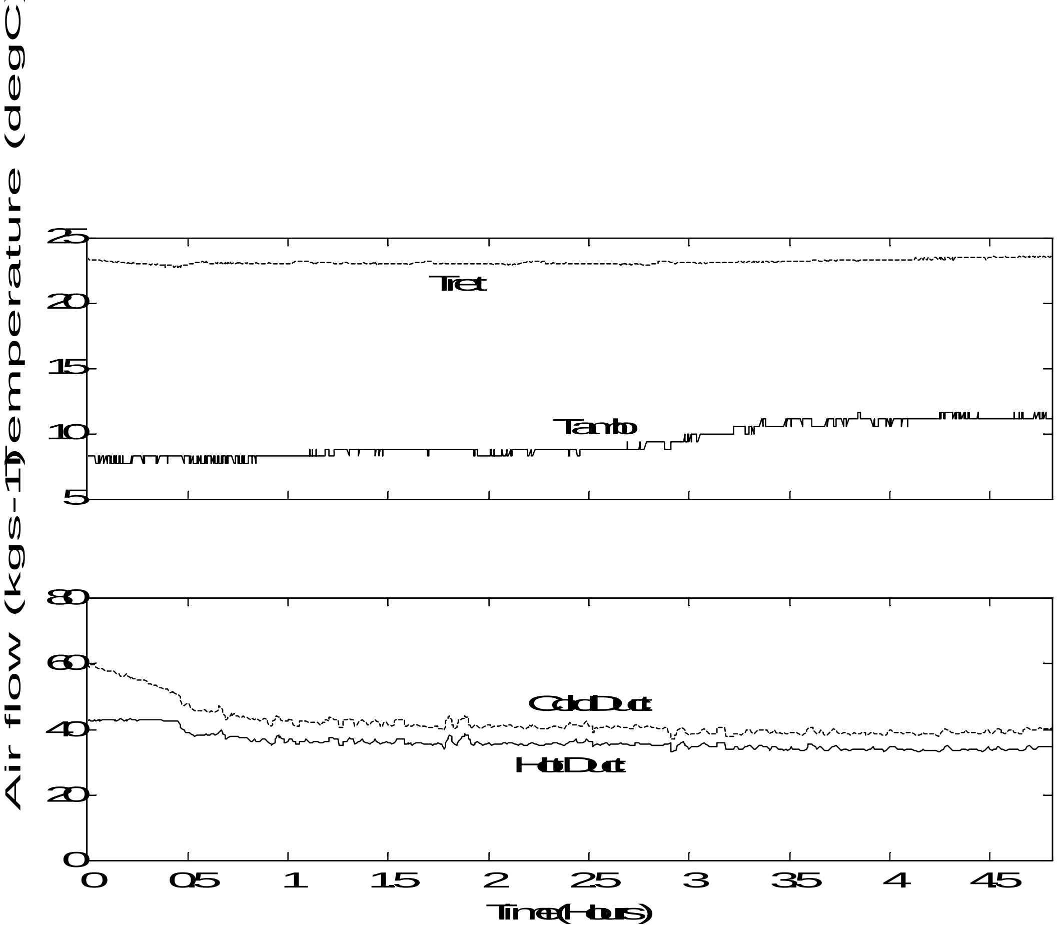


Figure 17. Measured Disturbances Affecting AHU Performance During Initial Test

Figure 18 and Figure 19 show the test results. The top graph in Figure 18 shows the controlled temperatures and their setpoints and the lower graph shows the control signals to each of the three subsystems. Figure 19 shows the control signal errors and the setpoint tracking errors in the upper two graphs and the fault detector indicators in the three lower graphs. The first feature to note from Figure 18 is that the controllers are unable to regulate at the setpoints very well, despite relatively constant measured disturbances and constant setpoints. The source of much of the instability appears to be the mixing box, which is cycling about its setpoint. This causes the mixed air temperature to vary, which in turn affects the load on the heating and cooling coils in their respective ducts downstream of the mixing process. The cooling coil reacts to the cycling in the mixing process with more extreme variations, causing the cooling valve to vary across its entire range. The disturbances in the mixing process influence the heating process to a lessor degree. However, the heating coil controller is still unable to regulate very well the controlled variable at the setpoint.

In Figure 19, operational problems in the AHU are evident with the indicator variables exceeding thresholds for sustained periods. Thresholds on the control signals and the controlled variables were set arbitrarily for this test and were thus not established empirically from training data. The control signal thresholds were set to 0.25 (25% of range) and the controlled variable thresholds to 2K. The cycling in the mixing process triggers an alarm due to the controlled variable being more than 2K outside of the setpoint for more than the half-hour time limit (P_{max} in Figure 2) set for the tests. Onsite inspection of the mixing process revealed that leakage existed through the return-air dampers and this contributes to the control signal error exceeding the threshold at certain times, particularly when the controller demanded full outside-air ($u=1$).

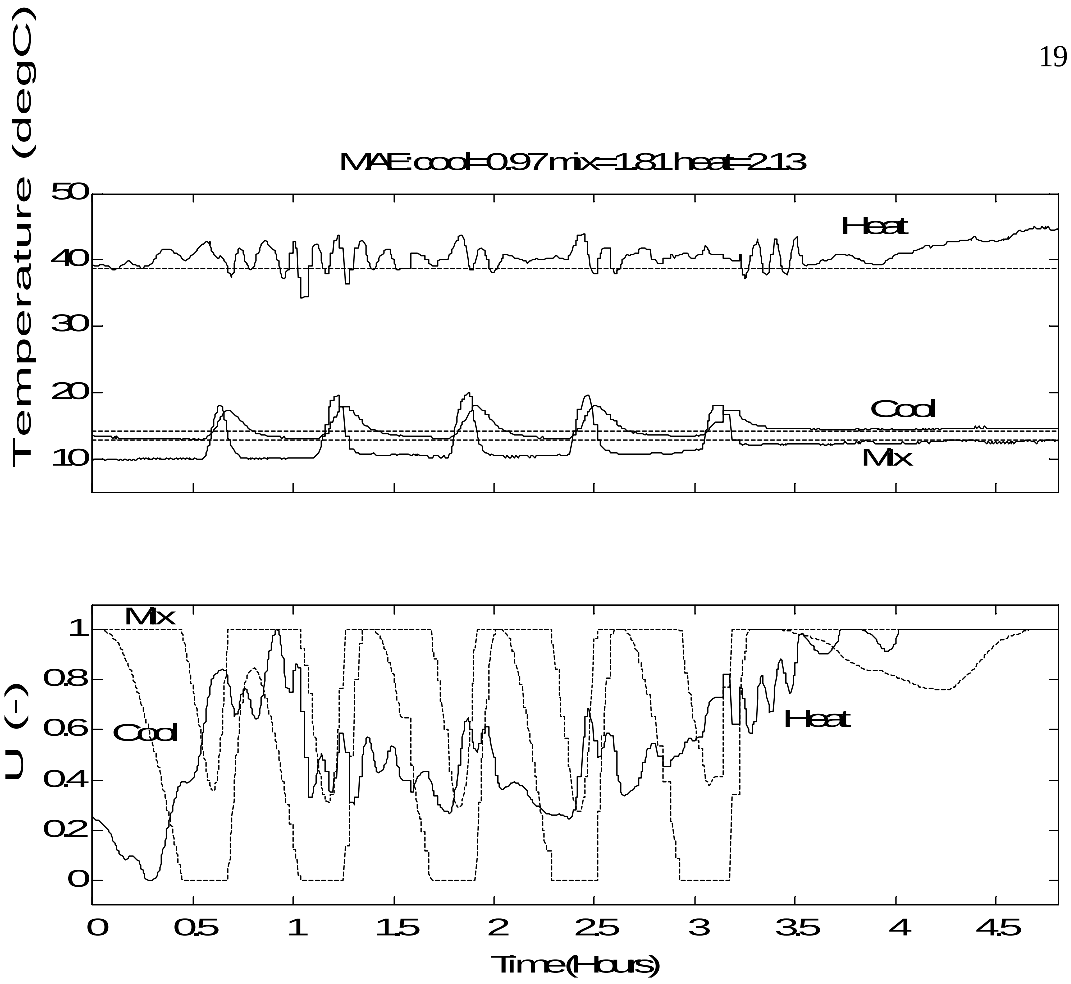


Figure 18. Control Performance

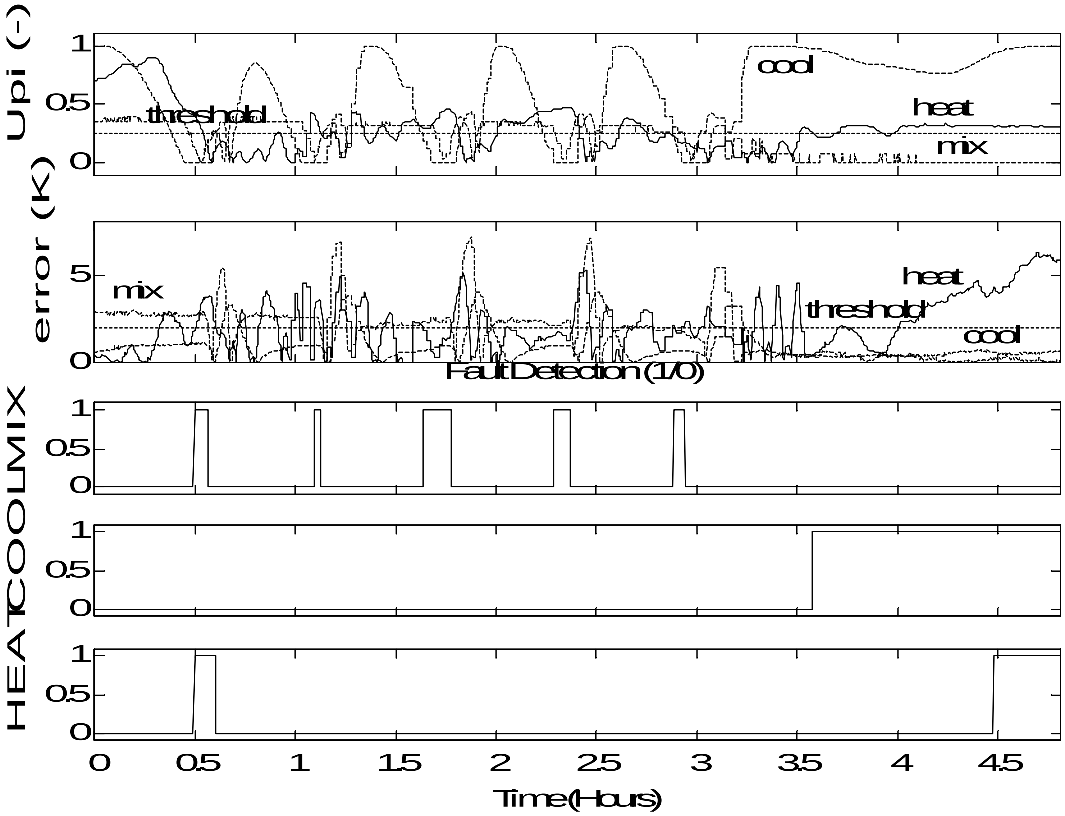


Figure 19. Fault Indicator Variables

Whenever the cooling process becomes active (i.e., the control signal is greater than zero), the error between the predicted and actual control signals is large, as shown in Figure 19. However, these large errors do not lead to an alarm, due to the cycling of the valve bringing the valve back to its closed position, where there is little prediction error before the half-hour time limit. The software thus only generates an alarm toward the end of the data when the coil valve stays open. The reason for the large error between the cooling control signal and the measured value was determined to be due to the chillers being disabled in the building during the test period. The cold water inlet temperature to the cooling coil was thus higher than expected since any cooling effect came only from the cooling towers. As the cold-water temperature is a *parameter* in the controller software and not a variable input, the models predict a greater degree of cooling than is actually produced. The software therefore demonstrates a capability for detecting faults in the primary plant systems.

There are two periods in the test data when the software generates alarms for the heating coil system. The first alarm instance is caused by a sustained error between the predicted and actual control signals. Examination of Figure 18 shows that in the period before the alarm, the heating valve is near or at its closed position while there is still a large difference in temperature across the coil. This behavior is inconsistent with the expectation of correct operation. The reason for the behavior is uncertain, but operators had reported leakage problems with the pneumatic valves on both the heating and cooling coils. The discrepancy between the predictions and measurements could thus be due to a large leakage through the valve. The second alarm instance is caused by simultaneous threshold transgressions in both the control signal and setpoint errors. The error between the controlled variable and the setpoint is quite significant as verified in Figure 18. It is possible that the simultaneous setpoint and control signal errors were due to a de-activation of the control loop, although we were unable to confirm this. The fact that the controller does not start to reduce the magnitude of the heating control signal as the setpoint error increases is strong evidence for a problem with the controller rather than with the heating process.

7 CONCLUSIONS

This paper has demonstrated the potential for using simplified simulation models as part of an HVAC control scheme. The simulation test results showed that the scheme was able to improve control performance and detect three different types of faults in a dual-duct air-handling unit. Results from the first phase of tests on a real AHU installed in a large office building demonstrated the fault detection capability of the control scheme and served to highlight practical implementation issues.

The performance of the control scheme and its ability to detect faults in the controlled process depends on the accuracy of the models. Realizable accuracy is affected by errors in the models, selection of model parameter values, and reliability of the sensor signals. The tests on the real AHU demonstrated the difficulty in establishing a baseline of “correct operation” with which to determine thresholds and validate the models. A decision thus has to be made at the time of establishing thresholds whether to accept observed behavior as being “correct” or to fix/tune the system to improve its performance. In the tests, the controller proved useful as a re-commissioning tool allowing detection of pre-existing faults such as leaking valves and dampers based on default thresholds. However, if these kind of faults were ignored by setting high threshold values the overall sensitivity of the tool would be reduced making new faults difficult to detect. The paper thus highlights the need for carrying out proper commissioning (and periodical re-commissioning) in order to ensure a fault-free starting condition upon the introduction of any fault detection device to an operational HVAC system.

8 ACKNOWLEDGEMENTS

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technology and Community Systems, and the Federal Energy Management Program, of the US Department of Energy under Contract No. DE-AC03-76SF00098.

9 REFERENCES

ASHRAE, 1995. “Standard 135-1995 - BACnet - A Data Communication Protocol for Building Automation and Control Networks (ANSI approved)”.

Clark, R. C. 1985. “HVACSIM+ Building Systems and Equipment Simulation Program Reference Manual”. Published by the U.S. Department of Commerce, Gaithersburg, MD 20899.

Gertler, J. 1988. “Survey of Model-Based Failure Detection and Isolation in Complex Plants”. IEEE Control Systems Magazine. Number 6. Volume 8. Page 3.

Glass, A. S., P. Gruber, M. Roos, and J. Tödtli. 1994. “Preliminary Evaluation of a Qualitative Model-Based Fault Detector for a Central Air-Handling Unit”. Proceedings of 3rd IEEE Conference on Control Applications, Glasgow.

Hepworth, S. J., A. L. Dexter. 1994. “Neural Network Control of a Non-Linear Heater Battery”. Building Services Engineering Research and Technology. Volume 15. Number 3. Page119.

Isermann, R. 1995. “Model-Based Fault Detection and Diagnosis Methods”. Proceedings of the American Control Conference, Seattle, Washington, USA. Page 1605.

Patton, R. J., J. Chen, S. B. Nielsen. 1995. “Model-Based Methods for Fault Diagnosis: Some Guidelines”. Transactions of the Institute of Measurement and Control. Volume 17. Number 2. Page 73.

Salsbury, T. I. 1998. “A Temperature Controller for VAV Air-Handling Units Based on Simplified Physical Models”. ASHRAE HVAC&R Research Journal. Volume 4. Number 3.